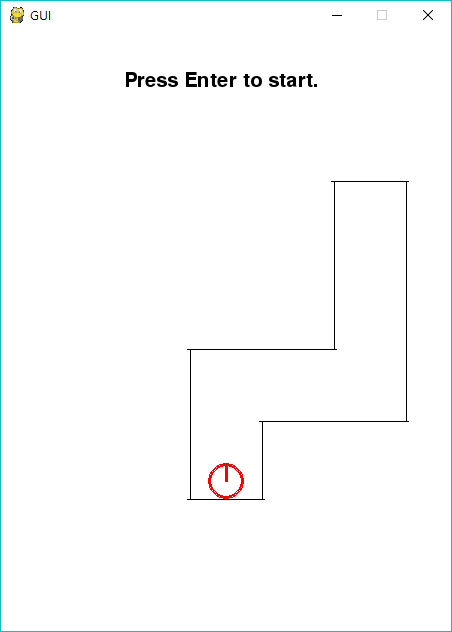
**計算型智慧**

**作業一**

**程式介面說明：**



* 按下Enter車子會開始動。
* 若碰觸到牆壁會直接關閉遊戲。
* 若車子中心進入終點區域，上面的Text會顯示”Arrived!”

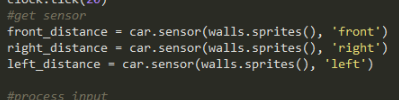
**程式碼說明：**

* 本次作業使用Python的Pygame完成

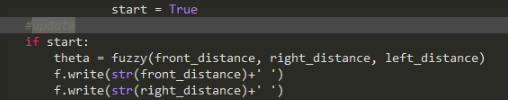
1. 利用Pandas將軌道讀入。
2. 建立Car與Wall的Class。
3. Car Class 裡面定義了get\_x(), get\_y(), get\_fi()與sensor的偵測。
4. Wall Class餵入牆壁資訊，並且畫出牆壁。
5. 利用Pygame內建的Function偵測碰撞。

* 程式執行的流程：

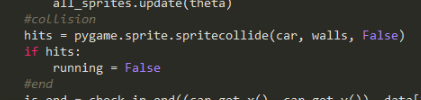
1. 回傳三個Sensor的值。



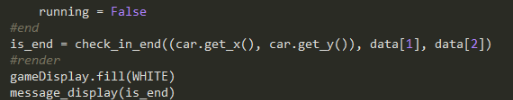
1. 將Sensor的值丟入模糊系統(fuzzy())
2. 經過前鑑步與後鑑步的運算，回傳方向盤應轉的角度，並且算出下一步的參數。



1. 判斷下一步的參數是否觸發撞牆。



1. 檢查車子是否到了終點區域。



**模糊規則設計：**

**規則：**

* IF Right is **LARGE** and Left is **SMALL**:

THEN turn\_right\_small

* IF Left is **LARGE** and Right is **SMALL**:

THEN turn\_left\_small

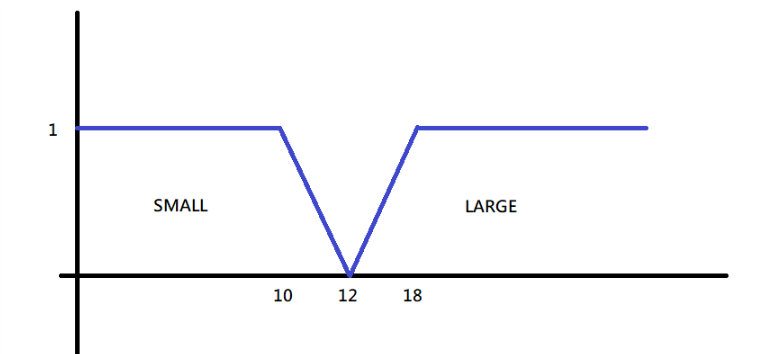
* IF Front is **SMALL** and Right is **LARGE**:

THEN turn\_right\_large

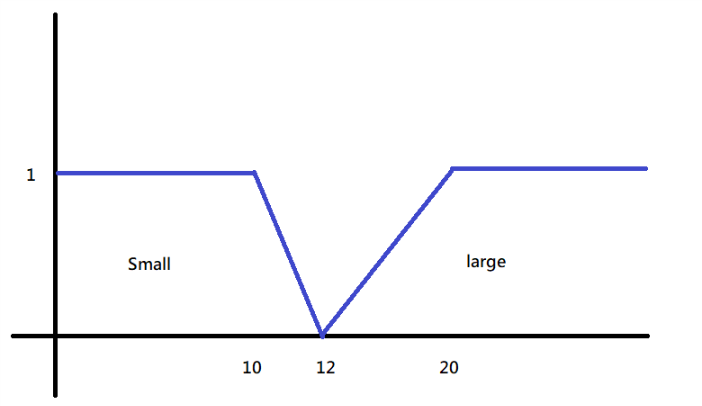
* IF Front is **SMALL** and Left is **LARGE**:

THEN turn\_left\_large

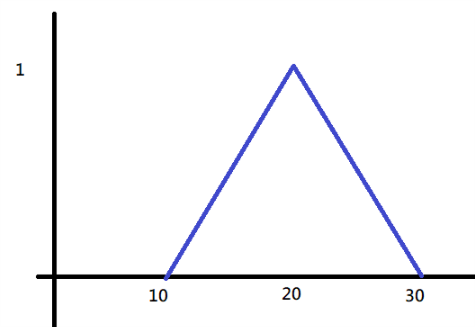
* **前面Sensor：**



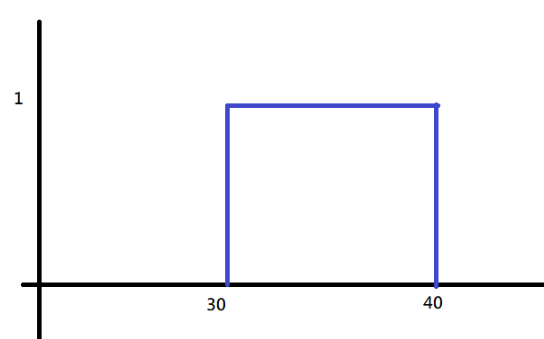
* **左右Sensor：**



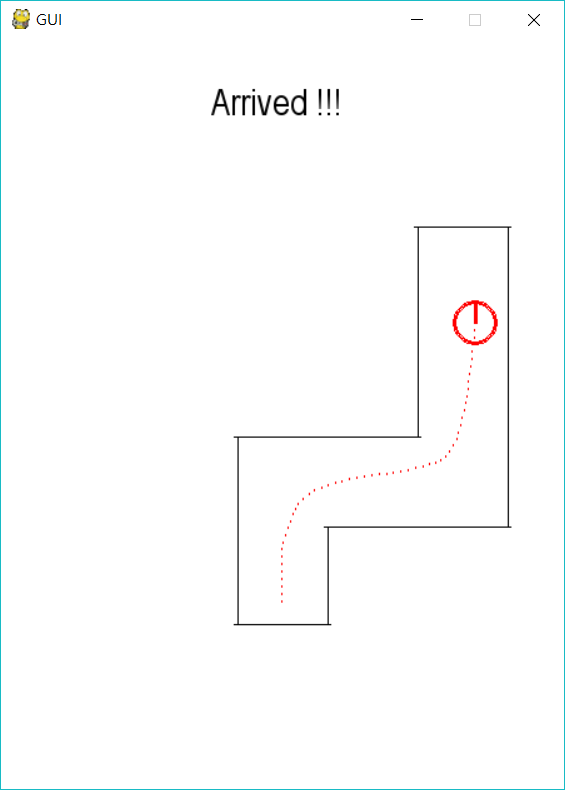
* **Turn Right/Left small:**



* **Turn Right/Left large:**



**實驗結果：**



* **分析：**

第一個彎過去之後，要盡量調整參數，將車身保持在軌道中間，車頭保持水平，若在第二個彎道出現時沒有調整好，很容易左邊就會碰觸牆壁，導致失敗。